Efficient Embeddings of Ternary Trees into Hypercubes

Ajay K. Gupta[1]
Donald Nelson[2]
{gupta, nelson}@cs.wmich.edu
Department of Computer Science
Western Michigan University
Kalamazoo, MI 49008

Hong Wang
hongwang@att.com
AT&T Network Services
Kansas, City, MO

[1]Research supported in part by a Fellowship from the Faculty Research and Creative Activities Support Fund (WMU-FRCASF 90-15), Department of Education (DOE R215K020362-622364479), and by the National Science Foundation under grants USE-90-52346, ACR-0000442 and ACI-0203776.

[2]Research supported in part by a grant from the WMU College of Arts and Sciences

Abstract

In this paper we present efficient graph embeddings for complete k-ary trees into boolean hypercubes. In particular we describe an efficient embedding of a complete ternary tree ($k = 3$) of height h into a hypercube, which achieves dilation 3 and expansion $\Theta(1.0104\ldots^h)$. The previously best known result is dilation 2 and expansion $\Theta(1.333\ldots^h)$. Our embedding achieves exponentially better expansion at the cost of an increase of 1 in the dilation. We also describe efficient embeddings of k-ary trees into hypercubes when $k = 2^p * 3^q$ for some $p, q > 0$ such that the embeddings achieve small constant dilation.

Keywords: embeddings, dilation, expansion, trees, hypercubes, emulation, algorithm porting, interconnection networks.

1 Introduction

The problem of embedding an n-node graph G into an n-node graph H is an important problem in parallel and distributed processing. Embedding results have been successfully used to establish equivalence of interconnections of parallel and distributed machines. Embeddings have also been successfully used to efficiently simulate an algorithm originally designed for a network G onto another network H. Embeddings and their implications to parallel processing have been studied extensively [1, 4, 7, 8, 11, 12]. In this paper we study the problem of embedding when the guest G is a complete k-ary tree, and H is a boolean hypercube.

The problem of efficiently embedding a complete k-ary tree into hypercube with $k \geq 3$ has largely remained unsolved, even though embeddings of complete and incomplete binary trees into hypercubes have been known for some time [2, 3]. In a more recent paper [12] the authors establish embeddings of complete k-ary trees into hypercubes and produce a function of k which bounds the dilation of the embedding. For the case $k = 3$ the dilation of their embedding is 2. In this paper we will produce an embedding with dilation 3 whose expansion, while not bounded, is exponentially smaller than the expansion of the embedding found in [12]. In addition, we will show embeddings for k-ary trees for values of k of the form $2^p, 3^p, 2^p * 3^q$ and $p, q > 0$. Before describing our results in detail, we define graph embeddings and their associated cost measures which measure the quality of an embedding.

An embedding $\phi =< f, g >$ of a guest G into a host H is defined by an injective mapping f from the nodes of G to the nodes of H together with a mapping g that maps every edge $e = (v, w)$ of G onto a path $g(e)$ connecting $f(v)$ and $f(w)$ in H. We refer to the mapping f as the assignment. Three commonly and extensively studied cost measures of an embedding are the dilation, the expansion, and the congestion [2, 7, 9]. The *dilation* δ is defined as the maximum distance between images in H when mapped under g between two adjacent nodes in G. The *expansion* ϵ is defined to be the ratio of the number of nodes in H to the number of nodes in G. The *congestion* of an edge e in H is the number of edges $e' = (u, v)$ in G such that the path $g(e')$ contains e. Expansion $\epsilon > 1$ implies that the host H is ϵ times larger than the guest G resulting in a 'waste' of the nodes in H which do not have a node assigned. Researchers have attempted to minimize dilation, expansion and/or congestion. Since most of the embeddings obtained in this paper achieve small constant congestion (3 in the case of ternary trees), for brevity we will not include the details for computing congestion.

Ideally we would like to obtain an embedding with $\delta = 1$ and $1 \leq \epsilon < 2$. Note that in an injective mapping f of the embedding ϕ expansion ϵ has to be at least 1. In general researchers have attempted to obtain $O(1)$-optimal embeddings in which a dilation δ of $O(1)$ and an expansion ϵ of $O(1)$ are achieved. An $O(1)$-optimal embedding of G into H implies that algorithms from G can be simulated by H so that the simulation incurs only a constant factor slow down and only a small fraction of nodes remain idle during the simulation. Furthermore, an $O(1)$-optimal embedding of G into H shows that computationally G and H are equivalent (within a constant factor).

Let $T^k(h)$ be a complete k-ary tree of height h. Every non-leaf node in the tree $T^k(h)$ has exactly k children, thus the tree has $(k^{h+1} - 1)/(k - 1)$ nodes. Furthermore, let $T^k(h)$ have $h + 1$ levels numbered $0, 1, \cdots, h$ such that the root of $T^k(h)$ is at level 0. Let $Q(d)$ be a d-dimensional boolean hypercube with 2^d nodes. Nodes in $Q(d)$ can be labeled by their binary representations $b_1 b_2 \cdots b_d$ where $b_i \in \{0, 1\}$. A node $b_1 b_2 \cdots b_{i-1} b_i b_{i+1} \cdots b_d$ of $Q(d)$ is connected in the i^{th} dimension to node $b_1 b_2 \cdots b_{i-1} \bar{b_i} b_{i+1} \cdots b_d$ where $\bar{b_i}$ indicates complement of b_i. Hence every node in $Q(d)$ is connected to exactly d other nodes. Observe that $Q(d)$ can be easily partitioned into two smaller hypercubes $Q(d-1)$ of dimension $d-1$ by simply removing all the edges in the i^{th} dimension for any fixed i, $1 \leq i \leq d$.

Our focus of this paper is to consider efficient embeddings of $(k^{h+1} - 1)/(k - 1)$-node complete k-ary trees $T^k(h)$ into $2^{d(h)}$-node hypercubes $Q(d(h))$ where $d(h)$ will be chosen to minimize expansion as much as possible by our strategy. The main result of this paper is an embedding of $T^3(h)$ into $Q(d(h))$ with dilation 3 and expansion that is $O(1.0105^h)$. This provides a significant improvement in the expansion over the embedding in [12]. The basic idea behind our embedding strategy is to start with an embedding of $T^3(h)$ into $Q(d(h))$ and construct the embedding of $T^3(h + 1)$ into $Q(d(h + 1))$ such that we have maximum possible size complete ternary trees that are composed of unassigned nodes in $Q(d(h + 1))$. (Of course, we have to have a complete ternary tree of height $h + 1$ in $Q(d(h + 1))$ which is composed of assigned nodes; *i.e.*, a mapping of $T^3(h + 1)$.) We then consider strategies for efficiently embedding $T^k(h)$ into $Q(d(h))$ for values of $k, (k > 3)$. Our main result here is that for $k = b2^p$ and $k = b3^p$ we can construct an efficient embedding ϕ with dilation $O(p + \delta_b)$, provided we are given a dilation δ_b embedding of $T^b(h)$ into a hypercube with

$O(1)$ expansion. This result allows us to more efficiently embed k-ary trees for at least the situations where $k = 2^p * 3^q$ for some $p, q > 0$.

Shen, *et al.* [12] also exhibit embeddings from $T^k(h)$ into $Q(d(h))$ where $d(h) = (h-1) * \lceil \log k \rceil + \lceil \log(k+2) \rceil$ for $k \geq 3$.[1] They establish the dilation $\delta(k) = max\{2, \psi(k), \psi(k+2)\}$ where ψ is a function that satisfies $\lfloor (\lceil \log(k) \rceil + 1)/2 \rfloor \leq \psi(k) \leq \lfloor \log k \rfloor$. Table 1 provides some comparisons between their embeddings and the embeddings in this paper for special values of k. As can be seen, all expansions grow exponentially with the height h of the tree. For the case $k = 3$ the base for the exponential growth of our result is significantly smaller (much closer to 1) than for that of Shen, *et al.* For instance, our expansion remains bounded by 2 for $h \leq 66$, while theirs does so only for $h \leq 2$. Similar results apply for the case $k = 3 \times 2^p$. On the other hand, the situation is not the same for $k = 3^q$ or $k = 3^q \times 2^p$. The exponent base in our expansion geometrically increases with q, exceeding 2 for $q \geq 67$, while the exponent base for their embedding remains bounded by 2. However, for $1 \leq q \leq 10$, ours compares as well or better, with equality at $q = 5$ and $q = 10$. In all cases their dilations are smaller except for $k = 2^p$ where the dilation is equal. The greatest differences occur for values of k containing a factor of 3^q.

The rest of the paper is organized as follows. In Section 2 we present the embeddings of $T^3(h)$. In Section 3 we consider embeddings for $T^k(h)$ for specific values of k, $k > 3$.

2 Embedding Complete Ternary Trees

In this section, we consider efficient embeddings of complete ternary trees into hypercubes. We present an embedding of a complete ternary tree $T(h)$ into a hypercube $Q(d)$ which achieves a dilation of 3 and $\Theta(1.0104\ldots^h)$ expansion.

Let $T(h)$ be a $(3^{h+1} - 1)/2$-node complete ternary tree of height h and let $Q(d)$ be a 2^d-node boolean hypercube of dimension d. There are at least two ways by which we can deduce embeddings of $T(h)$ into $Q(d)$. The first one results in a dilation of 3 and an expansion of $O(4^h/3^h)$ by first embedding $T(h)$ into a complete binary tree $B(2h)$ of height $2h$ [7] and then embedding $B(2h)$ into $Q(2h+1)$ [3]. The second one results in a dilation of $O(\log h)$ and optimal expansion by first embedding $T(h)$ into a complete binary

[1] Unless otherwise stated, all logarithms are base 2.

k	Shen, *et al.* [12].		This Paper	
	dilation δ	expansion	dilation	expansion
3	2	$\Theta((4/3)^h)$ $= \Theta((1.33\ldots)^h)$	3	$\Theta((2^{1.6}/3)^h)$ $= \Theta((1.0104\ldots)^h)$
3×2^p	$\frac{1}{2}(\log 3 + p)$ $\leq \delta \leq p+1$	$\Theta((1.33\ldots)^h)$	$p+4$	$\Theta((1.0104\ldots)^h)$
2^p	$p+1$	≤ 2	$p+1$	≤ 2
3^q	$(q \log 3)/2 \leq \delta$ $< q \log 3$	$\Theta((2^{\lceil q \log 3 \rceil}/3^q)^h)$	$3q$	$\Theta(((1.0104\ldots)^q)^h)$
$3^q \times 2^p$	$(p+1+(q \log 3))/2 \leq \delta$ $< p + q \log 3$	$\Theta((2^{\lceil q \log 3 \rceil}/3^q)^h)$	$p+3q+1$	$\Theta(((1.0104\ldots)^q)^h)$

Table 1: Embedding Comparisons

tree $B(\lceil \log(3^{h+1}-1) \rceil - 1)$ [2] and then embedding this complete binary tree optimally into hypercube [3]. Observe that while the first embedding achieves a smaller dilation with unbounded expansion, the second embedding achieves an unbounded dilation with optimal expansion. Shen *et al.* [12] have considered embeddings of $T^k(h)$ into hypercubes. For the case $k = 3$ the dilation of their embedding is 2 and the expansion is $\Theta((4/3)^h)$. In this section we describe an embedding of $T^3(h)$ into a hypercube which achieves a dilation of 3 and an expansion bounded by $O(1.0105^h)$.

In order to embed $T(h)$ into a hypercube with optimal expansion, the dimension d of the hypercube has to be $\lceil \log(3^{h+1}-1) - 1 \rceil$; however, as will be shown later we use dimension $d(h) = \lceil 1.6h \rceil + 1$.

We next describe our strategy for embedding $T(h)$ into $Q(d(h))$. The basic idea behind the strategy is as follows: Suppose we have an efficient embedding ϕ of $T(h)$ into $Q(d(h))$. In order to obtain an embedding of $T(h+1)$ into $Q(d(h+1))$, we first try to use as many unassigned nodes of $Q(d(h))$ as we can so that: (1) the dilation of the resulting embedding is the same as ϕ; (2) $Q(d(h+1))$ is at most $Q(d(h)+2)$; and (3) in the resulting embedding we have unassigned nodes which can be used to form embedded ternary trees with maximum possible height ($\leq h+1$). The novelty of our approach lies in the way we have unassigned and assigned nodes during the embedding of $T(h+1)$ into $Q(d(h+1))$ with dilation 3. Before describing the details of our embedding, we next define the notations that are used throughout the embedding.

Given a complete ternary tree $T(h)$ of height h, we denote the root as $t_0(h)$ and its three children as $t_1(h-1)$, $t_2(h-1)$, and $t_3(h-1)$. Continuing in this fashion, if a non-leaf node of $T(h)$ is labeled by $t_x(h')$ then its three children are labeled by $t_{x1}(h'-1)$, $t_{x2}(h'-1)$ and $t_{x3}(h'-1)$, for $0 \leq h' \leq h-1$. In an embedding of $T(h)$ into $Q(d(h))$, if node $t_x(h')$ of $T(h)$ is assigned to node w of $Q(d(h))$, then we may interchangeably refer to w as $t_x(h')$ and *vice versa*. Furthermore, we will select two specific nodes u and v such that u is at most distance two from $t_0(h)$, and v is adjacent to u. The particular choices of u and v will be critical to the embedding. Let $FR(h)$ denote a forest of complete ternary trees with maximum height h such that the trees represent the embeddings using the unassigned nodes of $Q(d(h))$ in an embedding of $T(h)$ into $Q(d(h))$. If there exist y complete ternary trees of height h', $h' \leq h$ and $y \geq 1$, in $FR(h)$, then we denote them as $F_1(h'), F_2(h'), \cdots F_y(h')$. For a free complete ternary tree $F_i(h')$, we denote its root as $f_i(h')$ and the root's three children as $f_{i1}(h'-1)$, $f_{i2}(h'-1)$, and $f_{i3}(h'-1)$. Continuing in this fashion, if a non-leaf node of $F_i(h')$ is labeled by $f_x(h'')$, $h'' \leq h'$, then its three children are labeled by $f_{x1}(h''-1)$, $f_{x2}(h''-1)$, and $f_{x3}(h''-1)$. Let $|T(h)|$ and $|FR(h)|$ denote the number of nodes in $T(h)$ and $FR(h)$, respectively.

We are now ready to describe our embedding of $T(h)$ into $Q(d(h))$ with dilation 3. Depending on the relative sizes of $T(h)$ and $Q(d(h))$ and whether $Q(d(h+1)) = Q(d(h)+1)$ or $Q(d(h+1)) = Q(d(h)+2)$, we have five cases. The basic strategy for each case is to start with an embedding of $T(h)$ into $Q(d(h))$ with dilation 3 and with a forest $FR(h)$ of unassigned nodes in $Q(d(h))$, and then extend this embedding so that we obtain an embedding of $T(h+1)$ into $Q(d(h+1))$ with dilation 3. Furthermore, we try to obtain as large a free forest $FR(h+1)$ as we can. Observe that if $Q(d(h+1)) = Q(d(h)+2)$, then we can obtain a larger forest $FR(h+1)$. Furthermore, in order to obtain the embedding of $T(h+1)$ into $Q(d(h+1))$, if $Q(d(h+1)) = Q(d(h)+1)$ then we have two copies of $Q(d(h))$ each containing an embedding of $T(h)$ with free forest $FR(h)$. Otherwise we have four copies of $Q(d(h))$ each containing an embedding of $T(h)$ with free forest $FR(h)$ since

$Q(d(h+1)) = Q(d(h)+2)$. We may label the i^{th} copy of $Q(d(h))$ in $Q(d(h)+2)$ or in $Q(d(h)+1)$ as $bin(i)Q(d(h))$, where $bin(i)$ is a binary representation of i with length 2 or 1 respectively, and $i = 0, 1, 2, 3$ or $i = 0, 1$. This in turn induces a labeling of all the nodes in $Q(d(h)+1)$ or $Q(d(h)+2)$. The result or output of each case will provide the input for the next case.

To provide the input or basis for Case 1, we show how to embed $T^3(1)$, $T^3(2)$, and $T^3(3)$ into hypercubes $Q(3)$, $Q(5)$, and $Q(6)$ respectively. Figure 1 shows a copy of $Q(6)$ divided into two copies of $Q(5)$, $(y = 0, 1)$. Each copy of $Q(5)$ consists of four copies of $Q(3)$, $(x = 00, 01, 10, 11)$. We will refer to nodes in this structure as $y/x/b_1b_2b_3$. Asterisks will be used to refer to sub-hypercubes. For example 0/0*/*** represents the top $Q(4)$ portion of the hypercube in Figure 1.

Setting $x = 00$ and $y = 0$ the embedding of $T^3(1)$ into $Q(3)$ is accomplished by assigning $t_0(1)$ to 0/00/000 and assigning its three children to 0/00/001, 0/00/010, and 0/00/011. This results in dilation and congestion 2. Note that the nodes are actually assigned to the hypercube 0/00/0**, and there is room for another copy of $T^3(1)$ in 0/00/1**.

Setting $y = 0$ and considering $Q(5)$ as four copies of $Q(3)$, the embedding of $T^3(2)$ is accomplished as follows. Assign the root $t_0(2)$ to 0/00/100. (Note that this node will be assigned $t_1(2)$ in the embedding of $T^3(3)$, as described in the next paragraph. For this embedding ignore that assignment.) The children $t_1(1)$, $t_2(1)$, and $t_3(1)$ of $t_0(2)$ are assigned to 0/00/000, 0/01/100, and 0/01/000 respectively. Assign the remaining nodes of these subtrees of height one in 0/00/0**, 0/01/1**, and 0/01/0** as was done for $T^3(1)$. Note that the node assignments all accur in 0/0*/***, leaving 0/1*/*** unassigned or free for another copy of $T^3(2)$. The dilation and congestion remain 2.

For the embedding of $T^3(3)$ into $Q(6)$, assign the root $t_0(3)$ to 0/00/101 and its three children, $t_1(2)$, $t_2(2)$, and $t_3(2)$, to 0/00/100, 1/00/100, and 1/10/100 respectively. The remaining nodes of these subtrees of height 2 are assigned in 0/0*/***, 1/0*/***, and 1/1*/*** as they were for $T^3(2)$. Note that the nodes in 0/1*/*** are unassigned and free for another copy of $T^3(2)$ rooted at 0/10/100. The dilation is 3, and the congestion is 2. Finally, observe that by setting u to 0/10/101 and v to 1/10/101, this embedding can be represented compactly by the diagram in Figure 2. Replacing 3 by h will provide the base case for our recursive embedding. We assume that $T^3(h)$ can be embedded in $Q(d(h))$ such that there

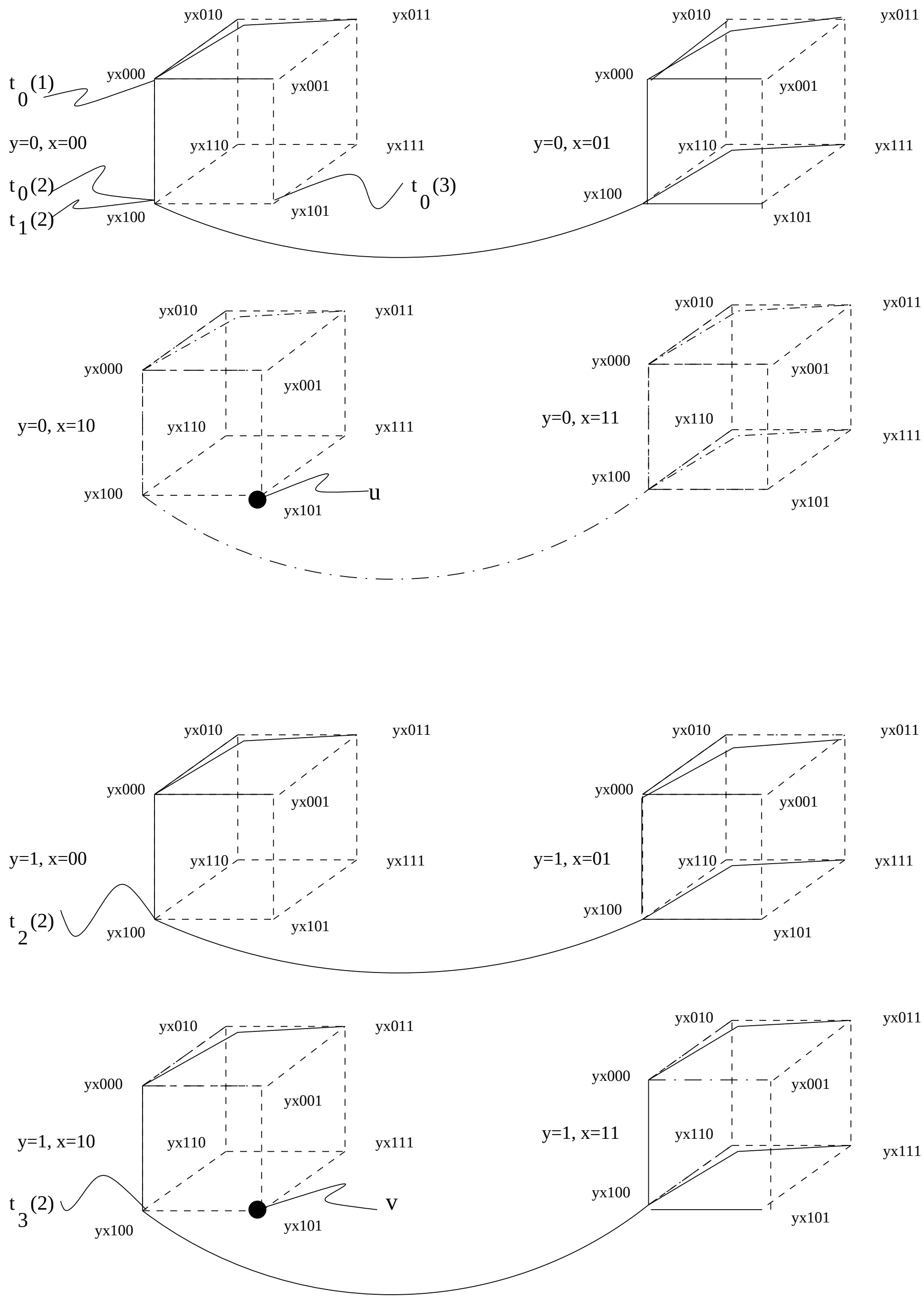

Figure 1: Embeddings of $T^3(1)$, $T^3(2)$, and $T^3(3)$.

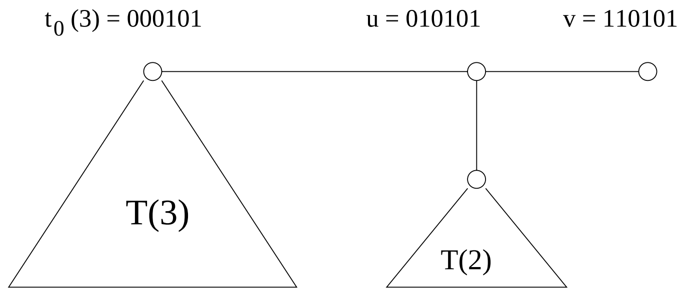

Figure 2: Representation for Embedding of $T^3(3)$.

is a free tree $FR(h-1)$ and additional free nodes u and v as illustrated. For clarity we will not show all connections in the diagrams that follow. The dilation of the embedding can be computed by considering the bit differences in the node assignments.

In the following cases, the notation $|FR(h)| \geq n|T(h')|$ will imply, in addition to the cardinality implications, that $FR(h)$ has room to embed n copies of $T(h')$ using the embedding strategy already defined for $T(h')$. The notation $|FR(h)| \geq n|T(h')| + 2$ will further imply that $FR(h)$ contains n copies of $T(h')$ plus two nodes u and v as described earlier. When node assignments are made in the figures and discussion, nodes on the left refer to those in the current case, and nodes on the right refer to those in the input or previous case. Finally, note that if $g(e)$ has dilation 3, there are several paths in the hypercube that could be chosen. In the embeddings that follow our goal is to keep the congestion at most 3. Part of the strategy for doing this will be to minimize the use of the edges connecting the new u and v to the adjacent embedded trees. In all but Case 4 these edges will have no congestion (*i.e.*, they are not used in the embedding) as they are supplied to the next case. This will maximize their availability in the subsequent embedding.

Case 1: $|FR(h)| \geq |T(h-1)| + 2$ and $d(h+1) = d(h) + 2$

We are given an embedding of $T(h)$ into $Q(d(h))$ with free forest $FR(h)$ in $Q(d(h))$ so that the forest contains a complete ternary tree $F_1(h-1)$ of height $h-1$ with dilation 3 and congestion at most 3. The root of $F_1(h-1)$ is adjacent to the unassigned node u of $FR(h)$, node u is adjacent to the root $t_0(h)$ of $T(h)$, and node v is adjacent to node u. We must construct the embedding of $T(h+1)$ into $Q(d(h+1))$ with free forest $FR(h+1)$ such

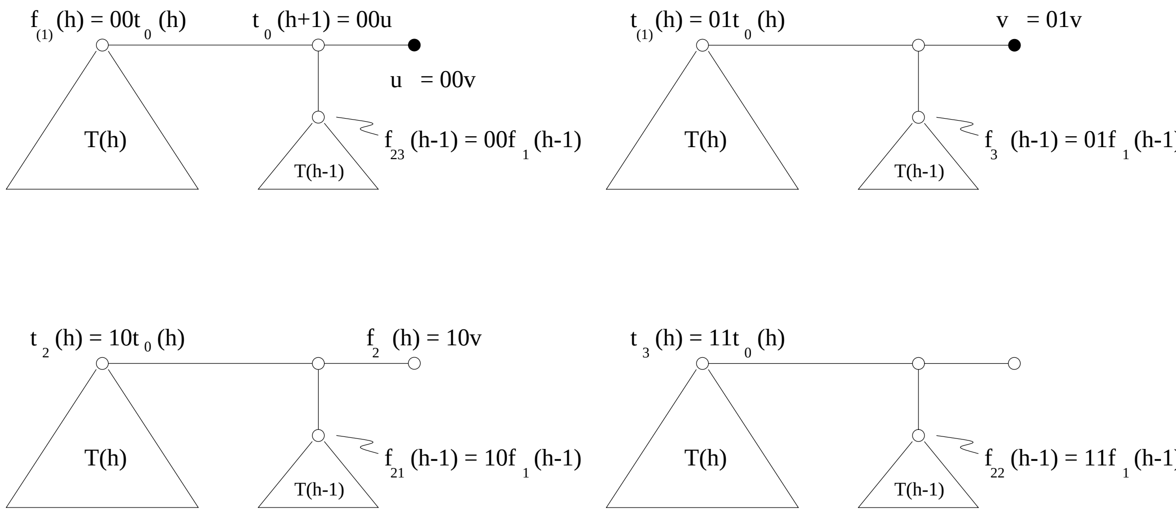

Figure 3: Sketch of an Embedding for Case 1.

that it has two complete ternary trees of height h, one complete ternary tree of height $h-1$, and two unassigned nodes u and v. Figure 3 presents one such embedding. Observe that we have four copies of $Q(d(h))$ into $Q(d(h+1))$. We assign node $t_0(h+1)$ to node $00u$, and let $t_1(h) = 01t_0(h)$, $t_2(h) = 10t_0(h)$ and $t_3(h) = 11t_0(h)$. Hence, we have the embedding of $T(h+1)$ in $Q(d(h+1))$ with dilation 3. In order to obtain the claimed free forest $FR(h+1)$, let $u = 00v$, $v = 01v$, and $f_1(h) = 00t_0(h)$ which gives us a free complete ternary tree $F_1(h)$. In order to construct the second free complete ternary tree $F_2(h)$ with dilation at most 3, let $f_2(h) = 10v$, $f_{21}(h-1) = 10f_1(h-1)$, $f_{22}(h-1) = 11f_1(h-1)$, and $f_{23}(h-1) = 00f_1(h-1)$. This completes the description of Case 1.

Case 2: $|FR(h)| \geq 2 * |T(h-1)| + |T(h-2)| + 2$ and $d(h+1) = d(h) + 1$

In this case, we have two copies of $Q(d(h))$ in $Q(d(h+1))$ such that every copy contains an embedding of $T(h)$ and free forest $FR(h)$ with dilation 3 and congestion at most 3. Our goal is to construct the embedding of $T(h+1)$ into $Q(d(h+1))$ so that the dilation remains 3 and we have a free forest $FR(h+1)$ containing one complete ternary tree of height $h-1$, two complete ternary trees each of height $h-2$, and two nodes u and v. Figure 4 presents one such embedding. Tree $T(h+1)$ can be viewed as node $t_0(h+1)$ connected to three complete ternary trees of height h with their roots as $t_1(h)$, $t_2(h)$ and $t_3(h)$. By assigning node

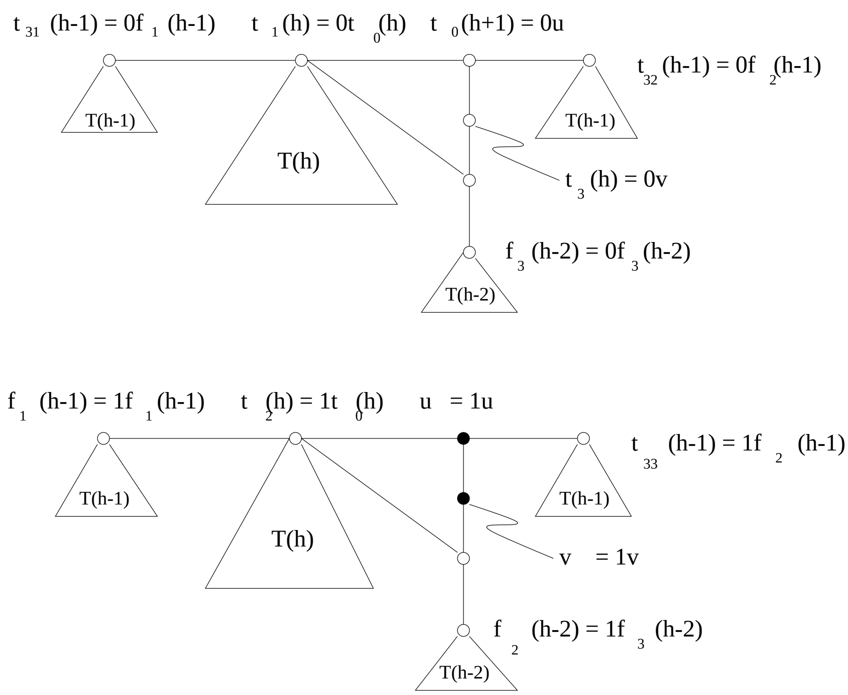

Figure 4: Sketch of an Embedding for Case 2.

$t_0(h+1)$ to node $0u$ and by letting $t_1(h) = 0t_0(h)$ and $t_2(h) = 1t_0(h)$, we can easily obtain the root of $T(h+1)$ and its first two complete ternary trees. In order to obtain the third complete ternary tree, we assign node $t_3(h)$ to free node $0v$ and let $t_{31}(h-1) = 0f_1(h-1)$, $t_{32}(h-1) = 0f_2(h-1)$ and $t_{33}(h-1) = 1f_2(h-1)$. It is not very hard to see that the dilation is kept at 3 and the congestion at most 3. In order to obtain the claimed free forest $FR(h+1)$, we let $f_1(h-1) = 1f_1(h-1)$, $f_2(h-2) = 1f_3(h-2)$, $f_3(h-2) = 0f_3(h-2)$, $u = 1u$ and $v = 1v$. It is again easy to see that the dilation and congestion for the three free complete ternary trees in $Q(d(h+1))$ are also 3.

Case 3: $|FR(h)| \geq |T(h-2)| + 2 * |T(h-3)| + 2$ and $d(h+1) = d(h) + 2$

In this case we are given a dilation 3 embedding of $T(h)$ with free forest $FR(h)$ in each of the four copies of $Q(d(h))$ in $Q(d(h+1))$. Our goal is to obtain a dilation 3 embedding of $T(h+1)$ into $Q(d(h+1))$ with free forest $FR(h+1)$ so that it contains one complete ternary tree $F_1(h)$ of height h, two complete ternary trees $F_2(h-1)$ and $F_3(h-1)$ of height $h-1$ each and free nodes u and v. Figure 5 presents one such embedding. The dilation 3 embedding of $T(h+1)$ can be easily obtained by assigning node $t_0(h+1)$ to node $00v$ and by letting $t_1(h) = 00t_0(h)$, $t_2(h) = 01t_0(h)$ and $t_3(h) = 10t_0(h)$. In order to obtain the free complete ternary tree $F_1(h)$ of height h, we simply let $f_1(h) = 11t_0(h)$. We can obtain the free tree $F_2(h-1)$ by assigning node $f_2(h-1)$ to node $00u$ and letting $f_{21}(h-2) = 00f_1(h-2)$, $f_{22}(h-2) = 01f_1(h-2)$ and $f_{23}(h-2) = 10f_1(h-2)$.

The free tree $F_3(h-1)$ can be viewed as node $f_3(h-1)$ connected to three complete ternary trees of height $h-2$ each with their roots as $f_{31}(h-2)$, $f_{32}(h-2)$ and $f_{33}(h-2)$. We further view the second complete ternary tree as having its root node $f_{32}(h-2)$ connected to three complete ternary trees of height $h-3$ rooted at nodes $f_{321}(h-3)$, $f_{322}(h-3)$ and $f_{323}(h-3)$. A similar thing holds for the third complete ternary tree of height $h-3$. In order to obtain $F_3(h-1)$ with dilation 3, we can assign node $f_3(h-1)$ to node $11v$, node $f_{32}(h-2)$ to node $01v$ and node $f_{33}(h-2)$ to node $10v$. Furthermore, we also assume $f_{31}(h-2) = 11f_1(h-2)$, $f_{321}(h-3) = 01f_2(h-3)$, $f_{322}(h-3) = 01f_3(h-3)$, $f_{323}(h-3) = 00f_2(h-3)$, $f_{331}(h-3) = 10f_2(h-3)$, $f_{332}(h-3) = 10f_3(h-3)$ and $f_{333}(h-3) = 11f_2(h-3)$. Finally, by letting $u = 01u$ and $v = 11u$ we have the claimed free forest $FR(h+1)$ with dilation and

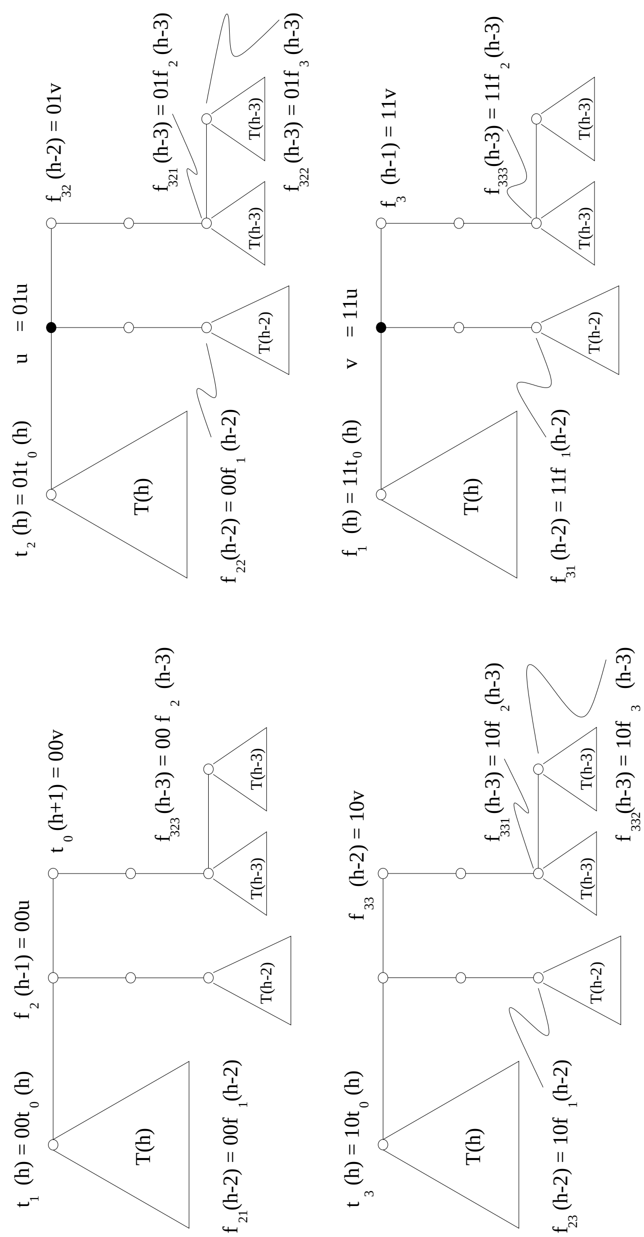

Figure 5: Sketch of an Embedding for Case 3.

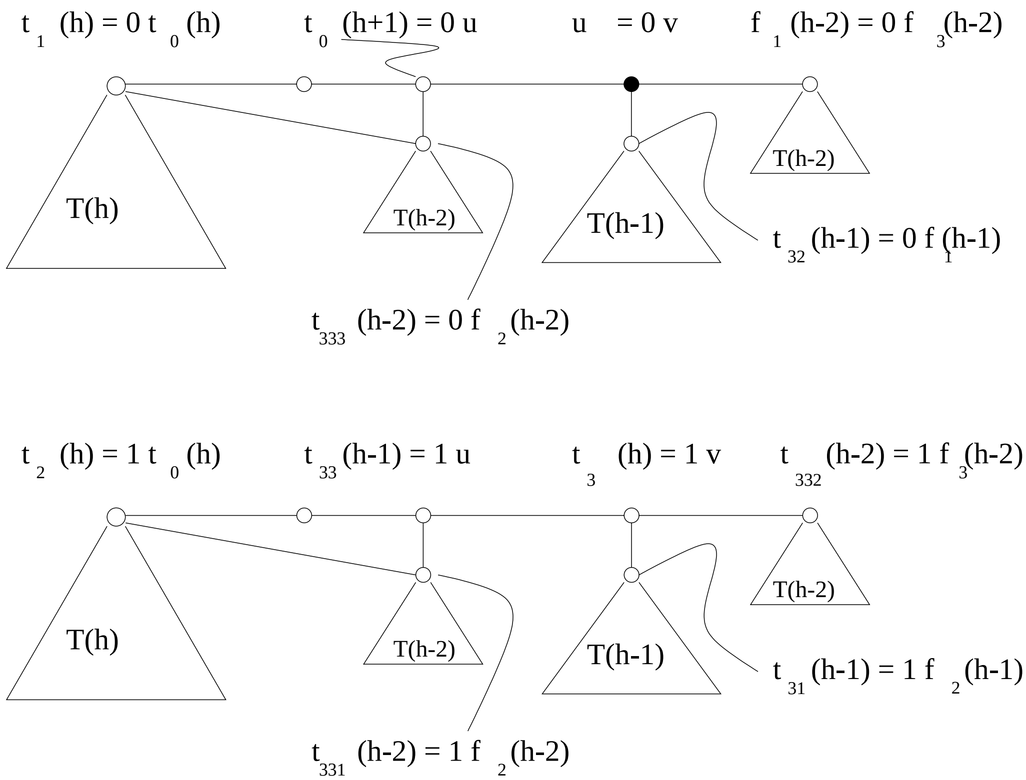

Figure 6: Sketch of an Embedding for Case 4.

congestion 3.

Case 4: $|FR(h)| \geq |T(h-1)| + 2 * |T(h-2)| + 2$ and $d(h+1) = d(h) + 1$

In this case we are given $Q(d(h+1))$ with two copies of $Q(d(h))$ such that each copy contains a dilation 3 embedding of $T(h)$ with free forest $FR(h)$. We are supposed to construct a dilation 3 embedding of $T(h+1)$ into $Q(d(h+1))$ with free forest $FR(h+1)$ so that it contains a complete ternary tree $F_1(h-2)$ of height $h-2$ and a free node u. Figure 6 presents one such embedding. As in some of the previous cases, we again view tree $T(h+1)$ to be composed of its root $t_0(h+1)$ which is connected to three complete ternary trees rooted at nodes $t_1(h)$, $t_2(h)$ and $t_3(h)$. We further view the third tree to be composed of its root $t_3(h)$ so that it is connected to three complete ternary trees with their roots as $t_{31}(h-1)$, $t_{32}(h-1)$ and $t_{33}(h-1)$, and subsequently the complete ternary tree rooted at $t_{33}(h-1)$ is also viewed to be further divided into its subtrees. As shown in Figure 6, by

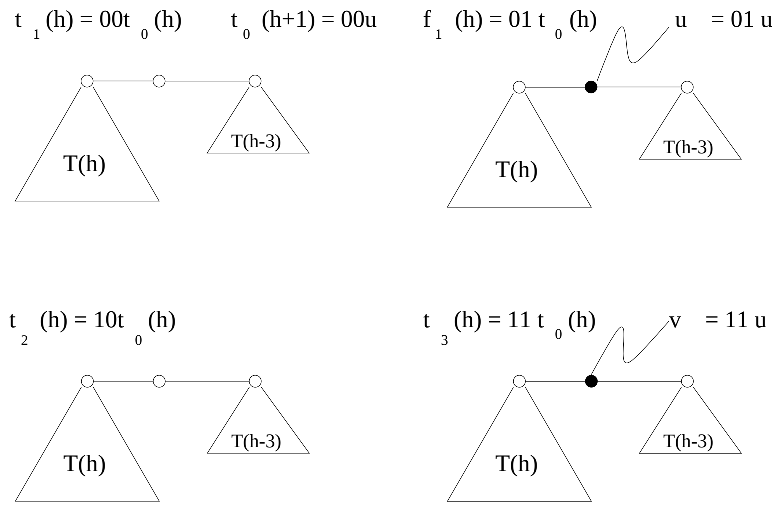

Figure 7: Sketch of an Embedding for Case 5.

assigning node $t_0(h+1)$ to node $0u$, node $t_3(h)$ to node $1v$, node $t_{33}(h-1)$ to node $1u$, and assuming $t_1(h) = 0t_0(h)$, $t_2(h) = 1t_0(h)$, $t_{31}(h-1) = 1f_1(h-1)$, $t_{32}(h-1) = 0f_1(h-1)$, $t_{331}(h-2) = 1f_2(h-2)$, $t_{332}(h-2) = 1f_3(h-2)$ and $t_{333}(h-2) = 0f_2(h-2)$, we can obtain a dilation 3 embedding of $T(h+1)$. Finally, the free forest $FR(h+1)$ can be easily obtained by assuming $f_1(h-2) = 0f_3(h-2)$ and $u = 0v$.

Case 5: $|FR(h)| \geq |T(h-3)| + 1$ and $d(h+1) = d(h) + 2$

Here, we are given four copies of $Q(d(h))$ into $Q(d(h+1))$ so that every copy contains a dilation and congestion 3 embedding of $T(h)$ into $Q(d(h))$ with free forest $FR(h)$. Our goal is to obtain a dilation 3 embedding of $T(h+1)$ into $Q(d(h+1))$ with free forest $FR(h+1)$ so that it contains a complete ternary tree $F_1(h)$ of height h with free nodes u and v. Figure 7 presents one such embedding. In order to obtain the dilation 3 embedding of $T(h+1)$, we assign node $t_0(h+1)$ to node $00u$ and let $t_1(h) = 00t_0(h)$, $t_2(h) = 10t_0(h)$ and $t_3(h) = 11t_0(h)$. In order to obtain the free tree $F_1(h)$ we use the free tree rooted at node $01t_0(h)$. Finally, to complete this case we let $u = 01u$ and $v = 11u$.

Now, observe that after Case 5 we have exactly the same conditions that are needed by Case 1 and hence our embedding can loop through cases $1, 2, 3, 4$, and 5 to obtain embeddings of $T(h)$ into $Q(d(h))$ for higher values of $h \geq 9$. The base case for input to Case 1 had congestion 2, and the congestion for Case 5 output was 3. It should be pointed out that the edges connecting u and v to the embedded tree and to the free tree in Case 5, are not used in the embedding itself; therefore, they have congestion zero when supplied to Case 1. This will provide sufficient path choices in subsequent embeddings to keep the congestion at 3. In each of these cases in order to obtain the embedding of $T(h+1)$ from the embedding of $T(h)$, we used the condition that when the height of ternary tree increases by 1, the number of nodes in the hypercube needs to at least increase by a factor of two but no more than a factor of four from the original size. Repeating cases 1 through 5 results in a repeating pattern of increases for $d(h)$ of $2, 1, 2, 1, 2$. Therefore, $d(h)$ can be described recursively as follows:

$$d(h) = \begin{cases} 3 & \text{if h = 1} \\ 5 & \text{if h = 2} \\ 6 & \text{if h = 3} \\ d(h-1)+2 & \text{if } h \bmod 5 = 1, 3, \text{ or } 4,\ h > 3 \\ d(h-1)+1 & \text{if } h \bmod 5 = 0 \text{ or } 2,\ h > 3 \end{cases}$$

Theorem 1 *Any complete ternary tree $T(h)$ of height h can be embedded into a hypercube $Q(d(h))$ of dimension $d(h) = \lceil 1.6h \rceil + 1$ with dilation and* 3.

Proof: We use mathematical induction to show that $d(h)$ as defined above equals $\lceil 1.6h \rceil + 1$. It is easily verified for $1 \leq h \leq 8$. Let $m \geq 9$ be an integer, and assume it is true for all values of h, $1 \leq h < m$. Note that $d(h+5) = d(h) + 8$ for $h \geq 3$. Therefore, $d(m) = d(m-5) + 8 = \lceil 1.6(m-5) \rceil + 1 + 8 = \lceil 1.6m - 8 \rceil + 9 = \lceil 1.6m \rceil + 9 - 8 = \lceil 1.6m \rceil + 1$. ∎

Theorem 1 gives us an embedding of an n-node tree $T(h)$ into an m-node $Q(d(h))$ so that the dilation and congestion are 3 and the expansion is $n/m = 2^{d(h)+1}/(3^{h+1} - 1)$. More specifically, the expansion used by our embedding is $\Theta(2^{1.6h}/3^h) = \Theta(1.0104\ldots^h)$, which remains bounded by 2 for $h \leq 66$. (A complete ternary tree of height 66 would have in excess of 10^{31} nodes!)

We conclude this section with Theorem 2 regarding the expansion of any embedding method which repeats in a manner similar to the repeating pattern of the embedding strategy above.

Theorem 2 *Let k be an integer greater than 2 which is not a power of 2, and let $\phi : T^k(h) \rightarrow Q(d(h))$ be an embedding strategy. For $h \geq 2$ let $\Delta(h) = d(h) - d(h-1)$. Suppose that there exist positive integers n and p such that if $h \geq n$ then $\Delta(h) = \Delta(h+p)$;* i.e., *the sequence of hypercube dimensions eventually increases in a finite repeating pattern. Then the embedding ϕ has exponential expansion.*

Proof: Let $m = \sum_{h=n}^{n+p-1} \Delta(h)$, the constant amount by which $d(h)$ increases over each block of length p. Over each such block the hypercube increases by a constant factor of 2^m, while the k-ary tree grows by a factor which is $\Theta(k^p)$. Since k is not a power of 2, $k^p \neq 2^m$. Therefore, $2^m > k^p$, since $Q(h)$ must remain large enough to contain $T^k(h)$ for all values of h. If we set $x = 2^{(m/p)}/k$, then $x > 1$, and the expansion of ϕ is $\Theta(x^h)$. ∎

If an embedding is to be a repeating type as described, it must have exponential expansion. It is, therefore, important to find repeating patterns that produce *small* exponent bases (as close to 1 as possible). Generally, it may prove too complex to expand this technique for other values of k, particularly for larger prime values.

3 Embedding Complete k-ary Trees

In the previous section we considered efficient embeddings of complete ternary trees into hypercubes. Naturally the question of efficient embeddings of complete k-ary trees with $k > 3$ into hypercube arises. In this section we provide results on such embeddings for specific values of k. Our main result of this section is the following: If there exists an efficient embedding of a complete b-ary tree $T^b(h)$ into a hypercube with dilation δ_b and expansion $O(f(h))$, $b \geq 2$, for some function f, then there exists an embedding of a complete k-ary tree into a hypercube with dilation $O(p + \delta_b)$ and expansion $O(f(h))$ whenever $k = b2^p \geq 2$. This result gives us efficient embeddings of complete k-ary trees whenever $k = 3 \times 2^p$ or $k = 2^{p'}$. In fact this result can also be generalized to obtain efficient embeddings for the situations when $k = b3^p$.

Let $T^k(h)$ be a complete k-ary tree of height h and $(k^{h+1} - 1)/(k-1)$ nodes. Let the root of $T^k(h)$ be at level 0, the root's children be at level $1, \cdots$, and leaf of $T^k(h)$ be at level h. We first describe some straight forward embeddings of $T^k(h)$ into hypercubes whenever k is a power of b for $b = 2$ or $b = 3$ so that they motivate our embedding strategies. We then describe our main result and its implications. Let us start with the problem of efficiently embedding a tree $T^k(h)$ into a hypercube $Q(d)$ for $k = 2^p \geq 2$. We know that $T^2(h)$ can be optimally embedded into $Q(h+1)$ with dilation 2 and expansion $O(1)$ [3]. Throughout this section, we refer to this embedding as BI-embedding. In order to embed $T^{2^p}(h)$ into $Q(d)$ we may use a two step embedding. In the first step we can embed $T^{2^p}(h)$ into a complete binary tree $T^2(ph)$ of height ph by simply assigning nodes at level i of $T^{2^p}(h)$ to nodes at level ip of $T^2(ph)$ in a 1-1 fashion, $0 \leq i \leq h$. It is easy to see that the dilation can be kept to be no more than p. In the second step we can now optimally embed $T^2(ph)$ into $Q(d) = Q(ph+1)$ using the BI-embedding. Combining the first and second step gives us the embedding of $T^{2^p}(h)$ into $Q(ph+1)$ with $O(p)$ dilation and $O(1)$ expansion. More precisely, the dilation is $p+1$ since only one edge of $T^2(ph)$ gets dilated by 2 in the BI-embedding.

Let us now consider the problem of efficiently embedding $T^{3^p}(h)$ into a hypercube. We can use a two step approach which is similar to the one used in the embedding of $T^{2^p}(h)$. In the first step, we can easily embed $T^{3^p}(h)$ into $T^3(ph)$ with dilation p by assigning nodes at level i of $T^{3^p}(h)$ to nodes at level ip of $T^3(ph)$ in a 1-1 fashion. In the second step, we can now embed $T^3(ph)$ into $Q(\lceil 1.6ph \rceil + 1)$ using the embedding of Section 2 with dilation 3. Combining these two steps gives us an embedding of $T^{3^p}(h)$ into $Q(\lceil 1.6ph \rceil + 1)$ with dilation $O(p)$ and expansion $O((2^{1.6}/3)^{ph})$. Note that if we use this two step approach, then the dilation has to be at least $\Omega(p)$ since the dilation of the embedding of $T^{3^p}(h)$ into $T^3(h)$ has to be $\Omega(p)$. We now return to the main result of this section, namely the one of efficiently embedding $T^{b2^p}(h)$ into a hypercube $Q(d)$.

Let there exist an embedding ϕ_b of $T^b(h)$ into $Q(d)$ with dilation δ_b and $|Q(d)| \geq |T^b(h)|$. Then, for $k = b2^p$ there also exists an embedding ϕ_k of $T^k(h)$ into $Q(ph+d)$ so that the dilation is $O(p+\delta_b)$. Observe that if embedding ϕ_b achieves an expansion that is $O(f(h))$, then embedding ϕ_k also achieves an $O(f(h))$ expansion. More specifically, if embedding ϕ_b uses hypercube $Q(d(h))$, then the embedding ϕ_k uses hypercube $Q(ph+d)$ and the expansion will remain $O(f(h))$. Theorem 3 establishes the existence of embedding ϕ_k. We note here that many straight forward embedding strategies result in dilation $O(p\delta_b)$ whereas our embedding ϕ_k achieves dilation $O(p+\delta_b)$. The basic idea for obtaining ϕ_k is to again use a two step approach, in which we first embed $T^{b2^p}(h)$ into an intermediate graph G that is composed of hypercubes and trees $T^b(h)$, and then in the second step we embed G efficiently into

$Q(ph+d)$. Lemma 1 establishes the result of the first step and Lemma 2 establishes the result of the second step. Throughout the rest of this section we use the following notations.

We assume $k = b2^p$ with $p \geq 1$ and $b \geq 2$ unless otherwise stated. Let $G = (V_G, E_G)$ and $H = (V_H, E_H)$ be two graphs. Then the graph product of G and H, denoted as $G \times H$, is defined (as in any standard graph theory text book) to be the graph with node set $V_G \times V_H$ in which node (u, v) is adjacent to node (u', v') if and only if either $u = u'$ and edge $vv' \in E_H$ or $v = v'$ and edge $uu' \in E_G$. We can also view $G \times H$ to be composed of $|V_G|$ copies of graph H such that the graph induced by the x^{th} nodes in these copies form the graph G for some fixed x (note that we choose x^{th} node from the first copy of H in $G \times H$, choose x^{th} node from the second copy of H and so on). Let $T^k(h)$ be a complete k-ary tree. We denote the root of $T^k(h)$ as $KR(h)$ and the i^{th} leaf of $T^k(h)$ as $KL_i(h)$, $1 \leq i \leq k^h$, where the leaves are ordered in a lexicographic manner. In a graph $Q(d) \times T^b(h)$, we have 2^d copies of $T^b(h)$ since hypercube $Q(d)$ contains 2^d nodes. We denote the root of the j^{th} copy of $T^b(h)$ in $Q(d) \times T^b(h)$ as $BR^j(h)$ and the i^{th} leaf in the j^{th} copy of $T^b(h)$ as $BL_i^j(h)$, $1 \leq i \leq b^h$ and $0 \leq j \leq 2^d - 1$. We next describe Lemma 1.

Lemma 1 *If $k = b2^p$, then any k-ary tree $T^k(h)$ can be embedded into $Q(ph) \times T^b(h)$ with dilation $p+1$ where "$\times$" is the graph product and $p \geq 1$.*

Proof: In order to describe the claimed embedding we use induction on h. For the basis case when $h = 1$, we would like to embed $(k+1)$-node tree $T^k(1)$ into a $((b+1)2^p)$-node graph $Q(p) \times T^b(1)$. In the graph $Q(p) \times T^b(1)$ we have 2^p copies of $T^b(1)$ labeled as copies $0, 1, \cdots, 2^p - 1$. By assigning node $KR(1)$ to node $BR^0(1)$ and leaf $KL_{(bi+j)}(1)$ to leaf node $BL_j^i(1)$ for $1 \leq j \leq b$ and $0 \leq i \leq 2^p - 1$, we have the embedding. It is also easy to see that the dilation is at most $p+1$ since $BR^j(1)$ is at most distance p away from $BR^0(1)$ (they belong to a hypercube of dimension p), and $BR^j(1)$ is adjacent to $BL_j^i(1)$. See Figure 8. Observe that leaves of $T^k(1)$ are assigned to leaves of $T^b(1)$'s in $Q(p) \times T^b(1)$.

Assume now by induction hypothesis that $T^k(h)$ can be embedded into $Q(ph) \times T^b(h)$ with dilation $p+1$ so that leaves of $T^k(h)$ are assigned to the leaves of $T^b(h)$'s in $Q(ph) \times T^b(h)$. We need to show that $T^k(h+1)$ can be embedded into $Q(p(h+1)) \times T^b(h+1)$ with dilation $p+1$ so that the leaves of $T^k(h+1)$ are assigned to the leaves of $T^b(h+1)$'s.

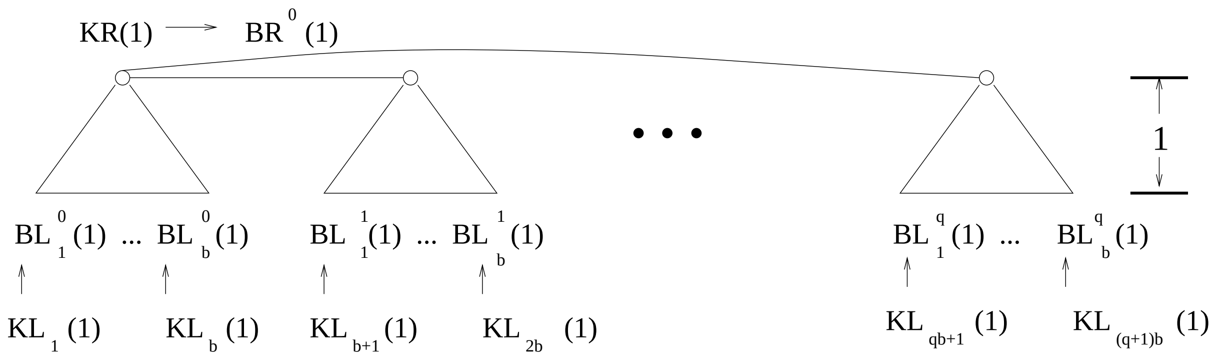

Figure 8: Embedding of Lemma 2 when $h = 1$. (Every triangle indicates $T^b(1)$. The assignment of the leaves of $T^k(1)$ is shown below the leaves of $T^b(1)$.)

We can view tree $T^k(h+1)$ as a tree $T^k(h)$ such that every leaf of $T^k(h)$ has k additional children which are also the leaves of $T^k(h+1)$. Furthermore, we view graph $Q(p(h+1)) \times T^b(h+1)$ to be composed of 2^{ph+p} copies of $T^b(h+1)$ such that they are labeled by trees i, j for $0 \leq i \leq 2^{ph}-1$ and $0 \leq j \leq 2^p - 1$. See Figure 9. We further view $T^b_{i,j}(h+1)$ as a tree $T^b_{i,j}(h)$ such that every leaf of $T^b_{i,j}(h)$ has additional b children which are also the leaves of $T^b_{i,j}(h+1)$. With this decomposition of graphs in mind, we can embed $T^k(h+1)$ into $Q(ph+p) \times T^b(h+1)$ as follows. We have the dilation $p+1$ embedding of $T^k(h)$ into $\bigcup_{i=0}^{2^{ph}-1} T^b_{i,0}(h)$ from the induction hypothesis since graph $\bigcup_{i=0}^{2^{ph}-1} T^b_{i,0}(h)$ is nothing but a graph $Q(ph) \times T^b(h)$. Furthermore, a leaf $KL_n(h)$ of $T^k(h)$ is assigned to a leaf $BL^{i,0}_m(h)$ for some i, n and m. Since nodes $BL^{0,j}_n(h)$ form a hypercube of dimension p for $0 \leq j \leq 2^p - 1$, we may assign the k children of $KL_n(h)$ in $T^k(h+1)$ to nodes that consist of the b children of $BL^{0,j}_n(h)$'s in $T^b_{0,j}(h+1)$ (a total of $b2^p = k$ children) such that the dilation between $KL_n(h)$ and its child is at most $p+1$. Using this idea in general, we can assign leaf $KL_x(h+1)$ of $T^k(h+1)$ to node $BL^z_y(h+1)$ of $Q(ph+p) \times T^b(h+1)$ such that integer $x = ((i_1 b^h + i_2)2^p + j)b + s$, integer $y = ib + s$ and $z = i_1, j$ (note z is a pair of two integers i_1 and j) with $0 \leq i_1 \leq 2^{ph} - 1$, $0 \leq i_2 \leq b^h$, $0 \leq j \leq 2^p - 1$ and $1 \leq s \leq b$. Since we are only 'extending' the embedding of $T^k(h)$ to obtain an embedding of $T^k(h+1)$ in such a way that every consecutive k/b leaves of $T^k(h+1)$ form a hypercube of dimension p and since the assignments of the leaves of $T^k(h)$ does not change from the original embedding (given by

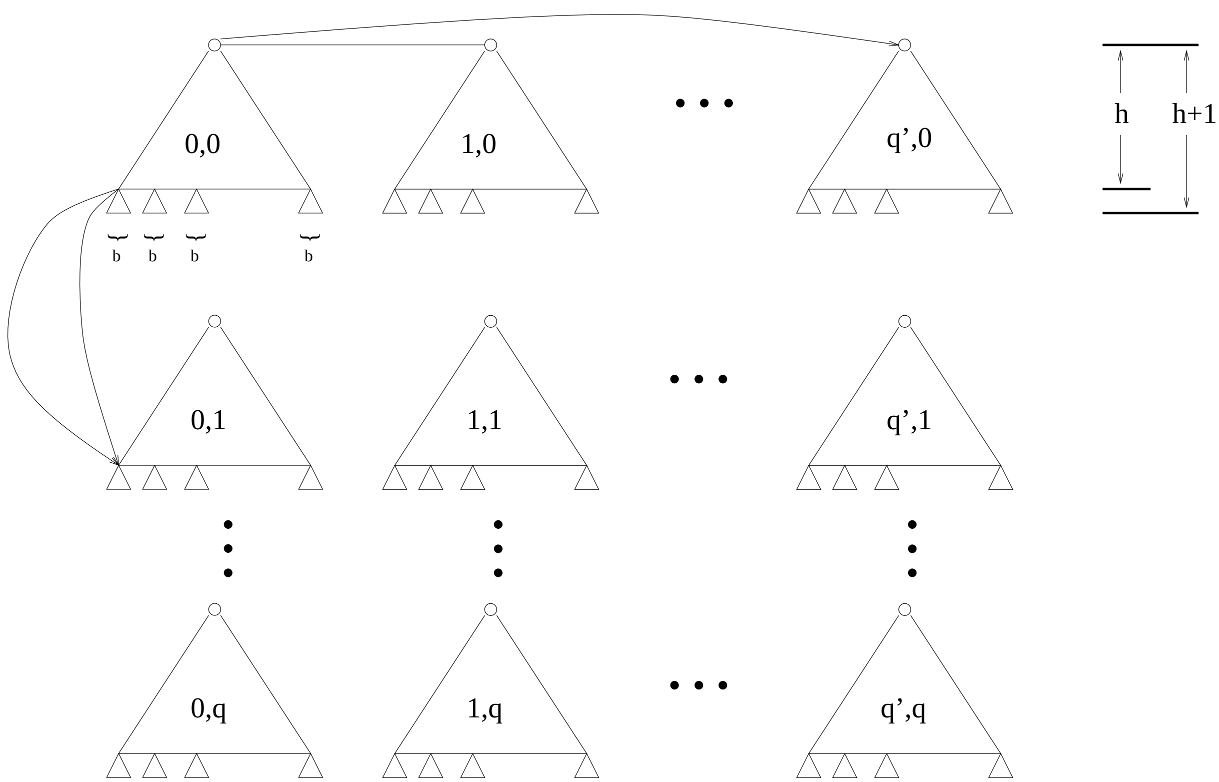

Figure 9: Embedding of Lemma 2 for the Induction Step. (Note $q = 2^p - 1$ and $q' = 2^{ph} - 1$. Every large triangle indicates a complete b-ary tree $T^b(h)$ and small triangle indicates $T^b(1)$.)

the induction hypothesis), it is easy to see that the dilation of the embedding of $T^k(h+1)$ into $Q(p(h+1)) \times T^b(h+1)$ remains at $p+1$. Furthermore, leaves of $T^k(h+1)$ are assigned to the leaves of $T^b(h+1)$'s in $Q(p(h+1)) \times T^b(h+1)$, and hence we complete the proof of this lemma. ∎

Lemma 2 *If $T^b(h)$ can be embedded into $Q(d)$ with dilation δ_b, then $Q(ph) \times T^b(h)$ can be embedded into $Q(ph+d)$ with dilation δ_b where "$\times$" is the graph product.*

Proof: The claim of the lemma follows trivially since $Q(ph+d)$ can be viewed as $Q(ph) \times Q(d)$ and we know that there exists an embedding of $T^b(h)$ into $Q(d)$ with dilation δ_b. ∎

Theorem 3 *If there exists an embedding ϕ_b of $T^b(h)$ into $Q(d)$ with dilation δ_b, then there exists an embedding ϕ_k of $T^{b2^p}(h)$ into $Q(ph+d)$ so that the dilation is at most $p+\delta_b+1$.*

Proof: By combining the results of Lemmas 1 and 2 and using the fact that only the edges of $T^b(h)$'s get dilated by δ_b during the embedding of $Q(ph) \times T^b(h)$ into $Q(ph+d)$, the theorem follows. ∎

Since our embedding of ternary trees achieves congestion 3, it is not hard to see that the embeddings of $T^k(h)$ for $k = b2^p$ achieve a congestion of $\Theta(\min\{b, 2^p\})$.

Corollary 4 *If the expansion of the embedding ϕ_b of Theorem 3 is $O(f(h))$, then the expansion of the embedding ϕ_k is also $O(f(h))$.*

We conclude this section by stating some of the implications of Theorem 3. The embedding strategy of Theorem 3 can be generalized to obtain efficient embeddings of complete $b3^p$-ary trees $T^{b3^p}(h)$ into hypercubes. We simply need to change the graph $Q(ph) \times T^b(h)$ of Lemma 1 to graph $Q(\lceil ph \log 3 \rceil + 1) \times T^b(h)$ and then use the ideas of the embedding of $T^{3^p}(h)$ into $Q(\lceil ph \log 3 \rceil + 1)$ to obtain an efficient embedding of $T^{b3^p}(h)$ into $Q(\lceil 1.6ph \rceil + 1) \times T^b(h)$. Theorem 3 along with Corollary 4 also give us the following corollaries which establish the result for efficiently embedding $T^{b2^p}(h)$ whenever $b = 2, 3$ and 3^q.

Corollary 5 *Any complete 2^p-ary tree $T^{2^p}(h)$ can be embedded into a $(ph+1)$-dimensional hypercube $Q(ph+1)$ with dilation $p+1$ and expansion $O(1)$.*

Corollary 6 *Any complete* $3*2^p$*-ary tree* $T^{3*2^p}(h)$ *can be embedded into a* $(ph+\lceil h\log 3\rceil+1)$*-dimensional hypercube* $Q(ph+\lceil 1.6h\rceil+1)$ *with dilation* $p+4$ *and expansion* $\Theta((1.0104\ldots)^h)$.

Corollary 7 *Any complete* 3^q*2^p*-ary tree* $T^{3^q*2^p}(h)$ *can be embedded into a* $(ph+\lceil qh\log 3\rceil+1)$*-dimensional hypercube* $Q(ph+\lceil 1.6qh\rceil+1)$ *with dilation* $p+3q+1$ *and expansion* $\Theta(((1.0104\ldots)^q)^h)$.

Corollary 8 *Any complete* 6^p*-ary tree* $T^{6^p}(h)$ *can be embedded into a* $(\lceil ph\log 6\rceil+1)$*-dimensional hypercube* $Q(\lceil 2.6ph\rceil+1)$ *with dilation* $4p+1$ *and expansion* $\Theta(((1.0104\ldots)^p)^h)$.

4 Conclusion

In this paper we have presented efficient embeddings of ternary trees into boolean hypercubes. We have also presented a few results on efficiently embedding complete k-ary trees into hypercubes for specific larger values of k.

Naturally, a number of questions regarding the embeddings of k-ary trees remain open. One of the very first questions which comes to mind is: Can the embedding strategy of a ternary tree be simplified further? The answer to this question may lead to efficient embeddings of k-ary trees when k is a larger prime number. Even though for most of the practical values of h our embedding of ternary trees achieves a small expansion, it is not optimal for all the values of h. Hence, the problem of obtaining embeddings of ternary trees into optimal size hypercubes remains open.

In this paper we considered only 1-1 embeddings of a k-ary tree into hypercube. If we allow many-to-one embeddings, can we obtain efficient embeddings for any value of k? Here, we would be more interested in the solutions where no more than $O(1)$ nodes of k-ary trees are assigned to a node of the hypercube.

5 Acknowledgements

The authors wish to thank the referees for their many valuable comments and suggestions, which have improved the presentation of the paper.

References

[1] F. Berman and L. Snyder. On mapping parallel algorithms into parallel architectures. *Journal of Parallel and Distributed Computing*, 4:439–458, 1987.

[2] S. Bhatt, F. Chung, F. T. Leighton, and A. Rosenberg. Optimal simulations of tree machines. In *the Proceedings of* 27^{th} *Symposium on Foundation of Computer Science*, pages 274–282, 1986.

[3] S. N. Bhatt and I. C. F. Ipsen. How to embed trees in hypercubes. Research Report 443, Dept. of Computer Science, Yale University, December 1985.

[4] M. Y. Chan and F. Y. L. Chin. On embedding rectangular grids in hypercubes. *IEEE Transactions on Computers*, 37(10):1285–1288, October 1988.

[5] A. K. Gupta. *On the Relationship between Parallel Computation and Graph Embeddings.* PhD thesis, Purdue University, August 1989.

[6] A. K. Gupta and S. E. Hambrusch. Load balanced tree embeddings. *Parallel Computing*, 18:595–614, 1992.

[7] J. W. Hong, K. Mehlhorn, and A. Rosenberg. Cost trade-offs in graph embeddings, with applications. *Journal of the Association of Computing Mahinery*, pages 709–728, 1983.

[8] R. Koch, T. Leighton, B. Maggs, S. Rao, and A. Rosenberg. Work-preserving emulations of fixed-connection networks. *Proceedings of 21st ACM Symposium on Theory of Computing*, 227–240, 1989.

[9] S. R. Kosaraju and M. Atallah. Optimal simulations between mesh connected array of processors. *Journal of the Association of Computing Machinery*, 35(3):635–650, July 1988.

[10] Arnold L. Rosenberg. Graph embeddings 1988: recent breakthroughs, new directions. In *In the Proceedings of the* 3^{rd} *Aegean Workshop on Computing, AWOC 88*, pages 160–169, June/July 1988.

[11] Y. Saad and M. H. Schultz. Topological properties of hypercubes. *IEEE Transactions on Computers*, 37(7):867–872, July 1988.

[12] X. Shen, Q. Hu, and W. Liang. Embedding K-ary complete tree into hypercubes. *Journal of Parallel and Distributed Computing*, 24:100-106, 1995.